

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. (Abdul Kadir, 2014:61).

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu. (Harianto Antonio dan Novi Safriadi, 2012:12).

Dari beberapa pengertian diatas, dapat disimpulkan bahwa sistem adalah sekumpulan elemen dari suatu prosedur kerja yang saling terhubung dalam melakukan suatu pekerjaan secara bersama-sama sehingga mencapai tujuan yang telah dibuat sebelumnya.

#### **II.1.1. Elemen Sistem**

##### **II.1.1.1. Tujuan Sistem**

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi motivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali. Tentu saja, tujuan antara

satu sistem dengan sistem lain berbeda-beda. Begitu pula yang berlaku pada sistem informasi. Setiap sistem informasi memiliki suatu tujuan, tetapi dengan tujuan yang berbeda-beda. Walaupun begitu, tujuan utama yang umum ada tiga macam, yaitu : (Abdul Kadir, 2014:62-63)

1. Untuk mendukung fungsi kepengurusan manajemen.
2. Untuk mendukung pengambilan keputusan manajemen.
3. Untuk mendukung kegiatan operasi perusahaan.

#### **II.1.1.2. Masukan Sistem**

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya, permintaan jasa dari pelanggan). Pada sistem informasi, masukan dapat berupa data transaksi, dan data non-transaksi (misalnya, surat pemberitahuan), serta instruksi. (Abdul Kadir, 2014:63).

#### **II.1.1.3. Proses Sistem**

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien. Pada sistem informasi, proses dapat berupa suatu tindakan yang

bermacam-macam. Meringkas data, melakukan perhitungan, dan mengurutkan data merupakan beberapa contoh proses. (Abdul Kadir, 2014:63).

#### **II.1.1.4. Keluaran Sistem**

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bias berupa suatu informasi, saran, cetakan laporan, dan sebagainya. (Abdul Kadir, 2014:63).

### **II.2. Informasi**

McFadden, dkk. (1999) Mendefenisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shannon dan Weaver, dua orang insinyur listrik, melakukan pendekatan secara matematis untuk mendefenisikan informasi. Menurut mereka, informasi adalah “jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima”. Artinya, dengan adanya sistem informasi, tingkat kepastian menjadi meningkat. Menurut Davis (1999), informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang. (Abdul Kadir, 2014:45).

Dari pengertian yang telah dijabarkan diatas, dapat ditarik kesimpulan bahwa informasi adalah data yang telah diproses sehingga menghasilkan sebuah pengetahuan dan dapat bermanfaat dalam pengambilan keputusan.

### II.3. Sistem Informasi

Ada beragam defenisi sistem informasi, sebagaimana tercantum di Tabel II.1 Berdasarkan berbagai defenisi tersebut, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan. (Abdul Kadir, 2014:8).

**Tabel II.1 Defenisi Sistem Informasi**

<b>Sumber</b>	<b>Defenisi</b>
Alter (1992)	Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.
Bodnar dan Hopwood (1993)	Sistem informasi adalah sekumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna.
Genilas, Oram, dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis computer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai.
Turban, McLean,	Sebuah sistem informasi mengumpulkan, memproses, menyimpan,

dan Wetherbe (1999)	menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
Wilkinson (1992)	Sistem informasi adalah kerangka kerja yang mengoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan ( <i>input</i> ) menjadi keluaran (informasi), guna mencapai sasaran – sasaran perusahaan.

(Sumber: Abdul Kadir, 2014:9)

Dari beberapa pengertian pada tabel II.1 diatas, bahwa sistem informasi dapat diartikan sebagai sekumpulan elemen prosedur kerja dari berbagai perangkat untuk mengolah informasi sehingga dapat mencapai tujuan atau sasaran suatu organisasi serta menghasilkan pengetahuan yang berguna bagi pengguna.

### II.3.1. Komponen Sistem Informasi

Sistem informasi mengandung komponen-komponen seperti berikut:

1. Perangkat keras (*hardware*), yang mencakup peranti-peranti fisik seperti komputer dan *printer*.
2. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.
4. Orang, yakni semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

5. Basis data (*database*), yaitu kumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai. (Abdul Kadir, 2014:71).

### **II.3.2. Klasifikasi Sistem Informasi**

Ada berbagai cara untuk mengelompokkan sistem informasi. Klasifikasi yang umum dipakai antara lain :

1. Level organisasi.
2. Area fungsional.
3. Dukungan yang diberikan.
4. Arsitektur sistem informasi.

Beberapa istilah sistem informasi lain juga sering dijumpai dalam literature, misalnya sistem informasi strategis dan sistem informasi geografis. (Abdul Kadir, 2014:89).

### **II.4. Sistem Pendukung Keputusan**

Pada dasarnya sistem pendukung keputusan merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif dimaksudkan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan seperti prosedur, kebijakan, teknik analisis, serta

pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan bersifat fleksibel.

Konsep Sistem Pendukung Keputusan (SPK) / Decision Support Sistem (DSS) pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah Management Decision Sistem. Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. (Desi Leha Kurniasih, 2013:7).

#### **II.4.1. Ciri-ciri Sistem Pendukung Keputusan**

Menurut Kosasi dan Kusriani (2007), adapun ciri-ciri sebuah SPK seperti yang dirumuskan oleh Alters Keen adalah sebagai berikut: (Desi Leha Kurniasih, 2013:7)

1. SPK ditujukan untuk membantu pengambilan keputusan-keputusan yang kurang terstruktur dan umumnya dihadapi oleh para manajer yang berada di tingkat puncak.
2. SPK merupakan gabungan antara kumpulan model kualitatif dan kumpulan data.
3. SPK memiliki fasilitas interaktif yang dapat mempermudah hubungan antara manusia dengan komputer.
4. SPK bersifat luwes dan dapat menyesuaikan dengan perubahan-perubahan yang terjadi.

#### **II.4.2. Karakteristik Sistem Pendukung Keputusan**

Karakteristik dari Sistem Pendukung Keputusan Yaitu: (Desi Leha Kurniasih, 2013:7)

1. Mendukung seluruh kegiatan organisasi.
2. Mendukung beberapa keputusan yang saling berinteraksi.
3. Dapat digunakan berulang kali dan bersifat konstan.
4. Terdapat dua komponen utama, yaitu data dan model.
5. Menggunakan baik data eksternal dan internal.
6. Memiliki kemampuan what-if analysis dan goal seeking analysis.
7. Menggunakan beberapa model kuantitatif.

#### **II.4.3. Kemampuan Sistem Pendukung Keputusan**

Kemampuan dari Sistem Pendukung Keputusan Yaitu: (Desi Leha Kurniasih, 2013:7)

1. Menunjang pembuatan keputusan manajemen dalam menangani masalah semi terstruktur dan tidak terstruktur.
2. Membantu manajer pada berbagai tingkatan manajemen, mulai dari manajemen tingkat atas sampai manajemen tingkat bawah.
3. Menunjang pembuatan keputusan secara kelompok maupun perorangan.
4. Menunjang pembuatan keputusan yang saling bergantung dan berurutan.
5. Menunjang tahap-tahap pembuatan keputusan antara lain *inteligensi*, *desain*, *choice*, dan *implementation*.

6. Kemampuan untuk melakukan adaptasi setiap saat dan bersifat fleksibel.
7. Kemudahan melakukan interaksi sistem.
8. Meningkatkan efektivitas dalam pembuatan keputusan daripada efisiensi.
9. Mudah dikembangkan oleh pemakai.
10. Kemampuan pemodelan dan analisis pembuatan keputusan.
11. Kemudahan melakukan pengaksesan berbagai sumber dan format data.

#### **II.4.4. Keterbatasan Sistem Pendukung Keputusan**

Keterbatasan dari Sistem Pendukung Keputusan, yaitu: (Desi Leha Kurniasih, 2013:7-8)

1. Ada beberapa kemampuan manajemen dan bakat manusia yang tidak dapat dimodelkan, sehingga model yang ada dalam sistem tidak semuanya mencerminkan persoalan sebenarnya.
2. Kemampuan suatu SPK terbatas pada pembendaharaan pengetahuan yang dimilikinya (pengetahuan dasar serta model dasar).
3. Proses-proses yang dapat dilakukan oleh SPK biasanya tergantung juga pada kemampuan perangkat lunak yang digunakannya.

SPK tidak memiliki kemampuan intuisi seperti yang dimiliki oleh manusia. Karena walau bagaimanapun canggihnya suatu SPK, hanyalah sautu kumpulan perangkat keras, perangkat lunak dan sistem operasi yang tidak dilengkapi dengan kemampuan berpikir. (Desi Leha Kurniasih, 2013:8).

#### **II.4.5. Tahapan Sistem Pengambilan Keputusan**

Menurut Herbert A. Simon, ada 4 tahap yang harus dilalui dalam proses pengambilan keputusan yaitu : (Desi Leha Kurniasih, 2013:8)

a. Penelusuran (intelligence)

Tahap ini merupakan tahap pendefinisian masalah serta identifikasi informasi yang dibutuhkan yang berkaitan dengan persoalan yang dihadapi serta keputusan yang akan diambil.

b. Perancangan (design)

Tahap ini merupakan tahap analisa dalam kaitan mencari atau merumuskan alternatif-alternatif pemecahan masalah.

c. Pemilihan (choice)

Memilih alternatif solusi yang diperkirakan paling sesuai.

d. Implementasi (implementation)

Tahap ini merupakan tahap pelaksanaan dari keputusan yang telah diambil. Metode Topsis.

#### **II.5. Jambu Madu**

Jambu air termasuk salah satu komoditas asli dari Indonesia. Salah satu jenis jambu air adalah jambu madu. Jambu madu merupakan salah satu jenis jambu air kualitas unggulan yang kita miliki.

A. Jenis-jenis Jambu Madu

Menurut Bapak Syahril, ada tiga jenis jambu madu, yaitu:

### 1. Jambu Madu Deli Hijau

Jambu madu Deli Hijau adalah jambu air madu yang berwarna hijau yang memiliki tingkat kemanisan yang luar biasa. Jambu madu hijau tidak memiliki biji di dalamnya sehingga membuatnya lebih nikmat saat dimakan. Perbedaan yang sangat mendasar dibandingkan dengan jambu madu lainnya adalah jambu madu Deli Hijau ini yang pertama, lebih cepat berbuah, jika bibit yang ditanam itu berumur 3-4 bulan mulai pencangkakan yang biasanya memiliki tinggi sekitar 60-70 cm, maka ia akan berbuah pada umur 8-9 bulan. Kemudian yang kedua adalah jauh lebih lebat buahnya walau memiliki ukuran yang jauh lebih kecil dibandingkan dengan jambu madu lainnya.

### 2. Jambu Madu Kesuma Merah

Jambu Madu Kesuma Merah adalah jambu madu air yang berwarna merah. Hal yang membedakannya dengan jambu madu lainnya adalah jambu ini tidak memiliki lubang biji di dalam buahnya yang artinya jambu ini dipenuhi dengan daging buah. Namun jambu kesuma merah ini memiliki waktu yang lebih lama berbuahnya, hal ini dikarenakan pohonnya yang besar dibandingkan dengan jambu madu deli hijau.

### 3. Jambu Madu Super Green

Jambu madu super merupakan jambu madu yang memiliki ukuran super besar dan warnanya hijau. besarnya jambu madu ini dapat mencapai ukuran satu botol aqua gelas, kalau ditimbang beratnya 1 kg 4 buah. Walau ukurannya besar namun tidak mengurangi rasa manis yang terdapat pada jambu madu ini. Jambu madu super ini memiliki warna yang hampir sama dengan warna jambu

madu deli hijau tetapi jambu madu super ini memiliki warna hijau kekuning-kuningan. Kemudian waktu berbuah jambu ini dapat disamakan dengan jambu madu Kesuma Merah yaitu waktunya sekitar 13-15 bulan berbuah, dan karena buahnya besar-besar maka jambunya juga tak selebat jambu madu Deli Hijau, tetapi karena pohonnya terbilag lebih besar maka jambunya juga dapat berbuah dengan banyak.

#### B. Ciri-ciri Jambu Madu Yang Baik

Menurut Bapak Syahril, ciri-ciri bibit jambu madu yang baik dapat dilihat dari uraian berikut:

##### 1. Bentuk Batang

Bentuk batang dari bibit jambu madu yang baik adalah batang yang besar dan tidak memiliki bekas luka atau juga cacat serta batang yang memiliki beberapa ranting yang besar.

##### 2. Bentuk Daun

Bentuk daun dari bibit jambu madu yang baik adalah dengan daun yang panjang dan juga lebar dan daunnya terlihat segar. Daun yang bagus berwarna hijau seperti daun-daun pada umumnya.

##### 3. Tinggi Tanaman

Tinggi tanaman bibit jambu yang baik dapat ditentukan dari ukurannya. Tinggi tanaman yang baik untuk sebagai bibit biasanya berkisar antara 60-70 cm.

## II.6. TOPSIS

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relative terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan. Hal ini disebabkan konsepnya sederhana, mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan. (Desi Leha Kurniasih, 2013:8).

Dalam sistem pendukung keputusan banyak metode yang bisa digunakan dalam melakukan perhitungan kriteria-kriteria dari suatu penelitian. Seperti AHP (analytic hierarchy process), SAW (Simple Additive Weighting), WP (Weighted Product), Fuzzy, Profile Matching, dan lain-lain. Dengan banyaknya metode yang ada dalam sistem pendukung keputusan, penulis hanya akan membahas tentang metode TOPSIS dalam penelitian ini.

### II.6.1. Langkah-langkah metode TOPSIS

Berikut adalah langkah-langkah dalam pengambilan keputusan menggunakan metode Topsis: (Desi Leha Kurniasih, 2013:8-9)

1. Membangun *normalized decision matrix* Elemen  $r_{ij}$  hasil dari normalisasi *decision matrix* R dengan metode *Euclidean length of a vector* adalah:

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{t=1}^m X_{it}^2}}$$

Dimana:

$r_{ij}$  = hasil dari normalisasi matriks keputusan R

$i = 1, 2, 3, \dots, m;$

$j = 1, 2, 3, \dots, n;$

2. Membangun *weighted normalized decision matrix* dengan bobot  $w = (w_1, w_2, \dots, w_n)$ , maka bobot normalisasi V adalah:

$$V = \begin{bmatrix} w_1 r_{11} & \dots & w_{1n} r_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} r_{m1} & \dots & w_{mn} r_{mn} \end{bmatrix}$$

3. Menentukan solusi ideal positif dan solusi ideal negative

Solusi ideal positif dinotasikan dengan  $A^+$  dan solusi ideal negative dinotasikan dengan  $A^-$  sebagai berikut:

Mementukan solusi ideal (+) dan (-)

$$A^+ = \{(\max V_{ij})(\min V_{ij} \mid j \in J), i=1,2,3,\dots,m\} = \{V_1^+, V_2^+, \dots, V_m^+\}$$

$$A^- = \{(\max V_{ij})(\min V_{ij} \mid j \in J), i=1,2,3,\dots,m\} = \{V_1^-, V_2^-, \dots, V_m^-\}$$

Dimana :

$V_{ij}$  = elemen matriks  $V$  baris ke- $i$  dan kolom ke- $j$

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ berhubungan dengan benefit criteria}\}$

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ berhubungan dengan cost kriteria}\}$

#### 4. Menghitung separasi

Separation measure ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negative.

Perhitungan matematisnya adalah sebagai berikut:

Separation measure untuk solusi ideal positif

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i=1,2,3,\dots,m$$

Dimana :

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan benefit criteria}\}$

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan cost kriteria}\}$

Separation measure untuk solusi ideal negatif

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \text{ dengan } i=1,2,3,\dots,m$$

Dimana :

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan benefit criteria}\}$

$J = \{j=1,2,3,\dots,n \text{ dan } j \text{ merupakan cost kriteria}\}$

5. Menghitung kedekatan relatif terhadap solusi ideal

Kedekatan relatif dari alternative  $A^+$  dengan solusi ideal  $A^-$  direpresentasikan dengan:

$$C_i = \frac{S_i^-}{S_i^- + S_i^+}, \text{ dengan } 0 < C_i^+ < 1 \text{ dan } i = 1, 2, 3, \dots, m$$

6. Meranking alternative

Alternatif dapat diranking berdasarkan urutan  $C_i^*$ . Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negative.

## II.7. *Unified Modeling Language (UML)*

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *UML* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

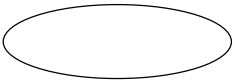
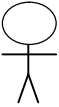
*UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015:93).



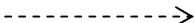
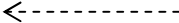
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis *UML* adalah sebagai berikut:

### 1. *Use case* Diagram

*Use case* diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini:

**Tabel II.2. Simbol *Use Case***

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem.</p> <p>Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.</p> <p>Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use</i></p>


	<i>case.</i>
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.



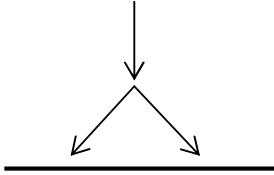
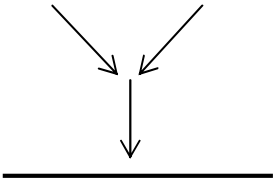
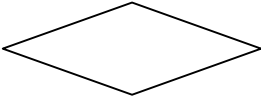

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015:94)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

**Tabel II.3. Simbol *Activity Diagram***

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.

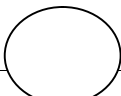
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

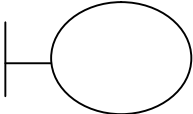
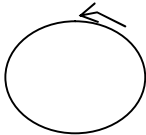

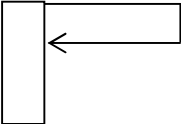


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015:94)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

**Tabel II.4. Simbol *Sequence Diagram***

<b>Gambar</b>	<b>Keterangan</b>
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi

	kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015:95)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 dibawah ini:

**Tabel II.5. *Multiplicity Class Diagram***

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar, 2015:95)

## **II.8. *Microsoft Visual Basic 2010***

*Microsoft Visual Basic 2010* adalah salah satu komponen *Microsoft Visual Studio 2010*. *Software* ini diluncurkan *Microsoft* pada tanggal 12 April 2010 dengan nama kode Dev10 dan menggunakan *.Net Framework 4.0*. *Integrated Development Environment (IDE)* pada *Visual studio 2010* telah didesain ulang sehingga lebih enak dipandang dan digunakan *programmer*.

Untuk *code* editor-nya, *Visual Basic* 2010 telah menambah fitur *highlights reference*. Ketika satu simbol / kode dalam bahasa pemrogramannya dipilih, maka simbol / kode yang sama, meskipun penggunaannya berbeda akan terlihat berwarna sama. Misal jika kode *math* dipilih, seluruh kode *math* akan terlihat berwarna sama. (Muhammad Saddam Naghfir, 2012:63-64).

## II.9. *SQL Server* 2008

*SQL Server* 2008 adalah sebuah *RDBMS* (*Relational Database Management System*) yang sangat powerful dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, *SQL Server* 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa *database*. *SQL Server* 2008 memiliki suatu *GUI* (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari-hari berkaitan dengan *database*, seperti menulis *T-SQL*, melakukan *backup* dan *restore database*, melakukan security *database* terhadap aplikasi, dan sebagainya. Pada *GUI* tersebut kita bisa melakukan settingan terhadap *SQL Server* untuk bekerja lebih optimal. Settingan juga bisa dilakukan menggunakan script untuk memudahkan developer mengubah *Setting Options* pada *SQL Server* 2008. (Ruslan, 2013:39).